NON-PROVISIONAL APPLICATION FOR UNITED STATES PATENT

Inventor(s):   Steven Paul Silverman
694 Harmony Orchard Rd.
Front Royal, VA 22630
United States citizen

Daniel Thomas Sullivan
13310 Cleveland Lane
Fort Washington, MD 20744
United States citizen

Invention: **MULTI-LEVEL EXPEDITED FORWARDING PER HOP BEHAVIOR**

078234/2

Multi-Level Expedited Forwarding Per Hop Behavior

CROSS REFERENCE TO RELATED APPLICATION

This application is based upon and claims benefit of copending and co-owned U.S.

Provisional Patent Application Serial No. 60/442,423 entitled *"Multi-Level Expedited*

5   *Forwarding Per Hop Behavior"*, filed with the U.S. Patent and Trademark Office on January 24,

2003 by the inventors herein, the specification of which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

The present invention relates to telecommunications technologies, and more particularly

10   to a method and system for supporting multiple levels of priority for voice or other types of

traffic over a telecommunications network.

BACKGROUND OF THE ART

Live voice communication is carried out over the Internet, and may thus be carried out

over other wide area networks, local area networks, or other distributed networks, pursuant to

15   Voice over Internet Protocol, or "VoIP." In using VoIP, a person's speech is digitized and

transmitted in Internet Protocol ("IP") packets from one VoIP phone to a second VoIP phone.

This may be implemented as a regular phone attached to an adaptor that converts analog signal to

VoIP. The caller's speech to the recipient is transmitted in one stream of packets, and the

recipient's speech is transmitted to the caller in a second stream of packets. Similarly, call setup

20   and teardown control signals are sent and received in IP packets.

Currently, VoIP speech and call control packets are transmitted in accordance with the

User Datagram Protocol (UDP), a communications protocol that offers a limited amount of

service when messages are exchanged between computers in a network that uses IP, but that does not provide sequencing of the packets that the data arrives in. UDP sends a packet from one IP address to a destination IP address and assumes the packet will arrive at the destination. UDP does not have guaranteed delivery of packets as the more commonly known Transmission

5   Control Protocol (TCP).

Under VoIP, voice packets between the caller and recipient are forwarded in the network by routers. A router may drop a packet or introduce uneven delays due to network congestion. Routers do not automatically re-route traffic to avoid congestion. As a result, a network router can drop call control or voice packets in VoIP phone calls. The effects of dropped VoIP packets

10   can prevent the establishment of VoIP calls or cause garbled speech being delivered to one of the parties in a phone call.

To avoid such congestion and to enable live voice communication to be successfully carried out over a wide area network, VoIP packets may be processed in a separate priority queue that the routers process first. Differentiated Services (also referred to as "DiffServ" or

15   "DS") offers a protocol that specifies and controls network traffic so that certain types of traffic get precedence. For example, in a network transmitting both voice and other data, voice data (which requires continuous, uninterrupted transmission) may take precedence over other kinds of data that are not sensitive to temporal interruptions in their transmission. Such other types of data packets (e.g., ftp, http, video, etc.) are thus processed in a separate queue commonly referred

20   to as the best effort queue. In order to provide the voice data packets with such differentiated service, the IP header of each voice data packet contains a Type of Service (ToS) byte, a portion of which is used to store a Differentiated Services Code Point (DSCP), a six bit value that

specifies a particular "per hop behavior" (PHB) for the data packet, i.e., a predefined forwarding behavior for the data packet.

One such PHB has come to be known as the expedited forwarding per hop behavior (EF PHB), as described in Davie, B. et al., "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246 (The Internet Society, March 2002), which is incorporated herein in its entirety by reference thereto. The EF PHB provides low loss, low delay, and low jitter VoIP services. The intent of the EF PHB is to provide a PHB in which packets marked for EF PHB ordinarily encounter short or empty queues, and maintain packet loss at a minimum by keeping the queues short relative to the available buffer space. Unfortunately, however, the EF PHB defines only one level of precedence and does not support multiple levels of precedence within the PHB. Thus, there currently exists a need for a multiple precedence level scheme to be applied to the EF PHB.

For example, during an emergency, such as a terrorist attack, severe weather, or other large-scale social calamity, there may be many people trying to make phone calls. With a VoIP telephone system, the increased volume of voice packets could exceed the bandwidth allocated by a router for VoIP. The result can lead to people being unable to place phone calls, or degrade existing calls to the extent that speech becomes unintelligible. If first-responders to an emergency are unable to establish critical communications, the results can be loss of life or significant loss of property.

As the commercial telephone sector is migrating toward a Packet Switching model rather than the Circuit Switching voice infrastructure that has been in use for many decades, the need is urgent to provide a VoIP system that will allow for such precedence based discrimination for voice traffic that is to be subjected to expedited forwarding.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for providing users preference in establishing and conducting VoIP calls by: (i) assigning precedence (a classmark) to calls which may be done by modifying the dialing plan; (ii) marking the actual data packets carrying the calls

5 so that they can receive special treatment; and (iii) giving right of way to the marked data packets in the routers of the distributed telecommunications system. One feature of this invention is the monitoring of the current size (the number of packets in the queue) of the priority output queue for each line. If the current queue size exceeds a preset limit for any traffic class, traffic in that class (or any lower class) is immediately dropped, thus saving space for

10 higher precedence traffic. Each higher traffic class is allowed a larger percentage of the maximum queue size than lower traffic classes.

To accomplish such multi-level discrimination among expedited forwarding traffic, the method and system of the invention utilize a multi-level expedited forwarding per hop behavior (MLEF PHB) that manages buffer space to provide right of way to calls based on a predefined

15 precedence scheme. The MLEF PHB defines a set of configurable parameters that define how much buffer space is available to each class or level of voice call. In a first exemplary embodiment of the invention, the MLEF PHB is implemented as programming statements in the kernel of a LINUX-based PC router. However, the PHB is independent of the particular router implementation, so the MLEF PHB can be applied to any router. Likewise, the PHB may

20 optionally be implemented as software, in an application specific integrated circuit (ASIC), or as programmed logic modules. There are various electronic devices of this type. The MLEF PHB applies an algorithm for dropping packets exceeding a caller precedence level-specific buffer

capacity, which variable buffer capacity is determined by the particular DSCP of the packet and the configuration of the router.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects, features, and advantages of the present invention will become more

5   apparent from the following detailed description of the preferred embodiment and certain modifications thereof when taken together with the accompanying drawings, in which:.

Fig. 1 is a table of address prefixes with associated precedence values according to the present invention;

Fig. 2 is a schematic view of steps to execute a VoIP phone call according to one

10   embodiment of the present invention;

Fig. 3 shows an example of an INVITE message sent from a caller user agent server to a proxy server;

Fig. 4 shows an example of an INVITE message sent from the proxy server to a receiver user agent server;

15   Fig. 5 shows an example of a RINGING message sent from the receiver user agent server to the proxy server;

Fig. 6 shows an example of a RINGING message sent from the proxy server to the caller user agent server;

Fig. 7 shows an example of a 200 OK response message sent from the receiver user agent

20   server to the proxy server;

Fig. 8 shows an example of a 200 OK response message sent from the proxy server to the caller user agent server;

Fig. 9 shows an example of an ACK message sent from a caller user agent server to a proxy server;

Fig. 10 shows an example of an ACK message sent from the proxy server to a receiver user agent server;

5 Fig. 11 shows an example of a BYE message sent from the receiver user agent server to the proxy server;

Fig. 12 shows an example of a BYE message sent from the proxy server to the caller user agent server;

Fig. 13 shows an example of a 200 OK message sent from a caller user agent server to a

10 proxy server;

Fig. 14 shows an example of a 200 OK message sent from the proxy server to a receiver user agent server;

Fig. 15 is a flowchart illustrating the process to receive and forward a VoIP packet according to one embodiment of the present invention;

15 Fig. 16 is a flowchart illustrating the process to enqueue data packets according to the present invention;

Fig. 17 is a flowchart illustrating the process to dequeue data packets according to the present invention;

Fig. 18 is a table of Type of Service values with maximum queue size for associated

20 precedence levels;

Fig. 19 shows an exemplary hardware system employing an embodiment of the present invention; and

Fig. 20 is an exemplary table of results from employing the present invention to resolve congestion in a priority queue.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The invention summarized above and defined by the enumerated claims may be better

5   understood by referring to the following description, which should be read in conjunction with the accompanying drawings in which like reference numerals are used for like parts. This description of an embodiment, set out below to enable one to build and use an implementation of the invention, is not intended to limit the enumerated claims, but to serve as a particular example thereof. Those skilled in the art should appreciate that they may readily use the conception and

10   specific embodiment disclosed as a basis for modifying or designing other methods and systems for carrying out the same purposes of the present invention. Those skilled in the art should also realize that such equivalent assemblies do not depart from the spirit and scope of the invention in its broadest form.

One aspect of the system and method of the instant invention is the MLEF PHB that

15   manages buffer space to provide right of way to high priority call traffic on a distributed network. The MLEF PHB defines a set of configurable parameters that define how much buffer space is available to each class of voice call within the low latency queue in a router. A configuration/initialization procedure calculates the maximum count in the buffer for each traffic class (MaxClassCnt (i)). This value is the maximum buffer percentage, i.e., the percentage of the

20   buffer that may be consumed by a particular class before packets begin to be dropped, times the buffer size. A maximum packet size may also be specified as part of the configuration. Using the standard G.711 voice encoding for VoIP, the bit rate is 64 kbps. At 50 packets/sec, each packet contains 160 bytes (1280 bits) of voice data. Each voice packet has 40 bytes of overhead

for the UDP, RTP, and IP headers. The maximum voice packet size is therefore 200 bytes per

packet (1600 bits) using the G.711 codec. (Secure voice would require a larger maximum packet

size since a copy of the IP header is encrypted to forestall certain attacks.) Once a maximum

packet size is set, the, buffer size for a line may be determined as follows:

5   1.   For a T1 line: (1.536 Mbps)*(.75)*(0.05 sec delay/pkt)/(1600 bits/pkt) = 36

packet buffer.

         2.   For an Ethernet link: 10Mbps*0.75*0.05 sec/1600 bits/pkt = 200 packet buffer.

The factor of .75 is used to reserve 25% of the bandwidth for other flows and routing

overhead. The factor of .05 seconds has been chosen as a reasonable maximum additional delay

10   for one hop on the path of a voice packet. (The ITU standard for total latency of a voice packet

is 150 ms.)

         As each packet is received by the router (or other component serving to buffer and

transmit the packet), any packet that is larger than the maximum packet size is dropped. The

current Buffer Occupancy Count (BOC), expressed as a number of packets (this is the number of

15   packets in the priority queue), is then checked against the threshold limit for the particular class

(MaxClassCnt(i)) and, if the BOC is less than the limit, the packet is enqueued on the MLEF

output queue. If the BOC is greater than or equal to this limit, the packet is discarded.

         It should be noted that all processing for the variable buffer management scheme may

optionally be performed at the byte level rather than at the packet level.

20   The following description sets forth an implementation of the MLEF PHB as described

above. First, an administrator must determine the number of classes in the system, assign a

single DSCP to each class, and determine the percentage of the buffer reserved for MLEF PHB

traffic that is to be used for each class and all traffic in a higher class (BP(i), where "i" is the

index value for the particular traffic class). The administrator must then set the maximum packet

size (MaxPacketSize) for all packets using the MLEF PHB, the total size of the buffer in a router

that is to be reserved for MLEF PHB traffic (MaxPacketCnt, calculated by the output bandwidth

of the router times 0.75 to leave a portion of the bandwidth available for router control), and, for

5   each class, the maximum number of packets that may be buffered for each class of traffic

(MaxClassCnt(i), calculated by MaxPacketCnt times BP(i) rounded up to an integer). For each

incoming VoIP packet, the packet size is first compared to the MaxPacketSize and, if the packet

size of the incoming packet is greater than the MaxPacketSize, such packet is discarded.

Otherwise, the Buffer Occupancy Count ("BOC," measured as the number of packets of MLEF

10   PHB traffic in the buffer) is determined and compared to MaxClassCnt(i). If the BOC is greater

than or equal to the MaxClassCnt(i) for the class to which the particular packet belongs, then the

packet is discarded. Otherwise, if the BOC is less than the MaxClassCnt(i) for the class to which

the particular packet belongs, then the packet is enqueued and the BOC value is incremented. In

forwarding the MLEF PHB traffic, the buffer dequeues each packet on a first-in first-out (FIFO)

15   basis, transmits each packet, and decrements the BOC value.

The above-described MLEF PHB may be implemented as programming statements or

hardware components in the output interface of a router. The following discussion sets forth an

exemplary embodiment of a system implementing the MLEF PHB.

First, call control is implemented through SIP (Session Initiation Protocol). When a high

20   priority user initiates a call, some way is needed to notify the network of the call's precedence.

Including the call precedence requires an additional feature in the call control protocol. A

Resource-Priority header field is thus added to the SIP message that identifies the precedence of

the phone call and is used by the SIP Proxy and user agents to determine the call precedence and

to set the Type of Service (ToS) value in call control and voice packets. A Resource-Priority header is necessary because the application program interface (API) for network sockets does not allow developers to determine the ToS value of an incoming packet. Without the Resource-Priority header, a developer would need to use kernel-level functions to determine the

5   precedence of an incoming packet. Inclusion of the call precedence value thus allows the SIP Proxy and user agents to assign the corresponding DSCP value to call control and voice packets.

The caller designates the precedence of the SIP phone call by issuing a prefix to the recipient's SIP address. The SIP user agent identifies the call precedence from the given SIP address prefix, adds the Resource-Priority header for the call precedence to the SIP message, and

10   sets the DSCP value, indicative of the ToS, for the call precedence in the SIP packet. During the phone conversation, the caller's and recipient's SIP phones set the DSCP value for the call precedence in the voice packets. Setting the DSCP value in the call control and voice packets is necessary for the router to prioritize the packets. In an exemplary embodiment of the system of the invention, open source components may be used to build the SIP user agents, softphone, and

15   SIP Proxy to implement the MLEF algorithm. The SIP agent provided in the exemplary embodiment of the system of the invention described herein may be a VoIP telephone. The SIP user agent sends SIP request and response messages to other user agents or to SIP proxies. A SIP proxy receives SIP messages from user agents and forwards the SIP messages to the destination SIP user agent. A common use of a SIP Proxy is to serve as a locator server. In the

20   exemplary embodiment described herein, the SIP Proxy is programmed to maintain state of the VoIP call in order to mark the Type of Service for the call control packets.

In order to initiate a telephone call using SIP, the caller dials the SIP address of the recipient. A SIP address is a uniform resource identifier (URI) with the format of

SIP:<destination user name>@<destination host><destination domain name>:<Port number>. If

a port number is not part of the URI, the port number is set at a default value of 5060. For

example, SIP:sally@sipserver.sallydomain.com would be the SIP address for a SIP phone with

the user name "sally" at SIP Proxy sipserver.sallydomain.com. The SIP Proxy will be listening

5   for messages on port 5060. In order for the caller to designate the call precedence, a prefix is

added to the recipient's username. The caller's user agent identifies the call precedence in the

destination SIP address. The caller's user agent then builds the SIP request message. When the

SIP user agent transmits the SIP request, the user agent sets the Differentiated Services field in

the ToS byte of the packet to the respective value for the call precedence (This field is typically

10   referred to as the Differentiated Services Code Point, DSCP). The SIP Proxy receives the SIP

request, and forwards the request to the destination user agent with the message ToS value set to

the call precedence. Responses from the recipient's user agent are sent to the caller via the SIP

Proxy.

Of course, the above call setup process is exemplary only, and in fact, call setup may be

15   carried out through alternate protocols, such as H.323 or MGCP, without departing from the

spirit and scope of the invention.

Figure 1 depicts an exemplary table correlating the SIP call prefix, Resource-Priority

header value, call precedence, DiffServ value, and ToS byte value. The DiffServ value

comprises the first six bits of the ToS byte and is referred to herein as the DSCP value. Of

20   course, the values provided in the table of Figure 1 may be assigned as the administrator deems

appropriate. However, of note is the fact that a different SIP Call Prefix is assigned to each

distinct Call Precedence, and associated with each distinct SIP Call Prefix is a unique DSCP

Value that sets the particular MLEF PHB to be applied to packets transmitted during the call.

The API for network sockets allows developers to directly set the ToS byte, which includes the DiffServ field. In the exemplary embodiment described herein, Differentiated Services is used in implementing MLEF, and the ToS byte value is used to set the precedence value of a packet using the network socket API.

5   Figure 2 is a schematic view of the steps taken by a caller using a SIP softphone to call a recipient. As depicted in Figure 2, the recipient is denoted as a "message robot" which is configured to stream a pre-recorded message to the caller. It should be noted, however, that such configuration was implemented for testing purposes only, and that, in practice, the caller may be any recipient equipped with any VoIP phone capable of communicating with the user agent and

10   SIP proxy to transfer VoIP data between the user agent and the recipient, whether they are located on the same or an entirely disparate network, and if the SIP user agent marks the ToS value in packets in accordance with the call precedence level. The user agent identifies the call precedence level in outgoing calls by matching the call precedence level pattern in the recipient's URI. The SIP user agent identifies the call precedence level in incoming calls using the value of

15   the Resource-Priority header in the incoming SIP call setup messages. Of course, as mentioned above, protocols other than SIP (such as H.323 and MGCP) may likewise be used to setup the call.

At step (1) of the exemplary embodiment, both the caller and the recipient have registered with the SIP Proxy. The caller's User Agent sends an INVITE message to the SIP

20   Proxy. The INVITE message contains the parameters necessary to conduct a voice conversation such as the caller's transport protocol, IP address, port number, and available coder-decoders (codecs). The SIP Proxy determines that it has the IP address and port of the recipient's User Agent and forwards the INVITE to the recipient's User Agent.

To further illustrate the concepts set forth herein, and by way of example only, SIP messages for a telephone call with a precedence level of Priority will be shown flowing from the softphone 1 to the message robot recipient through the SIP Proxy, as shown in Figure 2. To make a Priority call, the caller enters the recipient's SIP URI preceded by '*92*' to indicate that

5   this will be a Priority call, as follows: sip:*92*robot@sipserver.mobilesrv.com:5060. The caller's user agent then sets the INVITE packet ToS value to 180 (Figure 1) and sends the message to the SIP Proxy at IP address 192.168.1.234 and port 5060. The SIP message from the caller's user agent to the SIP Proxy is shown in Figure 3.

     The SIP Proxy receives the INVITE message from the caller and identifies that the caller

10   is requesting a Priority precedence from the prefix in the recipient's URI. The SIP Proxy adds a Resource Priority header with the value of "dsn.priority" to the INVITE message, establishes the Type of Service by setting the DiffServ value to 45, and sends the INVITE request to the recipient at 192.168.1.234 and port 5001. The INVITE message sent to the message robot is shown in Figure 4.

15   At step 2, the recipient's User Agent begins alerting the recipient of the incoming call and returns a "180 Ringing" response to the caller's User Agent. In the exemplary construction depicted in Figure 2, the recipient is a message robot and immediately sends the "180 Ringing" response upon receipt of the INVITE request. For purposes of this example, the message robot's User Agent sent the response message depicted in Figure 5 to the SIP Proxy at IP address

20   192.168.1.234 and port 5060 with the packet DSCP value set to 45. (This equates to the ToS byte value of 180.)

The SIP Proxy sets the DSCP value to 45 for a Priority packet and forwards the "180 Ringing" response to the caller's user agent at 192.168.1.102 and port 5060 as shown in Figure 6.

At step 3, when the recipient accepts the call, the recipient's User Agent returns a "200 OK" response with the IP address and port number to send the Real Time Protocol (RTP) voice packets to. In the exemplary embodiment described herein, the recipient immediately accepts the call and sends the "200 OK" response to the SIP Proxy. For purposes of this example, the recipient's User Agent sends the "200 OK" response message shown in Figure 7 to the SIP Proxy at IP address 192.168.1.234 and port 5060 with the packet DSCP value set to 45.

The SIP Proxy sets the DSCP value to 45 for a Priority packet and forwards the "200 OK" response to the caller's user agent at 192.168.2.102 and port 5060 as shown in Figure 8.

At step 4, when the caller's User Agent receives the "200 OK" from the recipient, the caller sends an ACK message to the recipient. ACK is used to acknowledge the reception of a final response to an INVITE. For purposes of this exemplary embodiment, the caller's user agent sets the ACK packet DSCP value to 45 and sends this message to the SIP Proxy at IP address 192.168.1.234 and port 5060. The ACK message from the caller's user agent to the SIP Proxy is shown in Figure 9.

The SIP Proxy then receives the ACK message from the caller and identifies that the caller is requesting a Priority precedence from the prefix in the recipient's URI. The SIP Proxy adds a Resource Priority header with the value of "dsn.priority" to the ACK message, sets the DSCP value to 45, and sends the ACK response to the recipient. The ACK message sent to the recipient is displayed in Figure 10.

At step 5, the caller and recipient conduct their phone conversation. In the exemplary

embodiment described herein, all the voice packets have their DSCP value set to the precedence

of the call requested in the INVITE message. The message robot recipient, in this example,

streamed a pre-recorded voice message using the Real Time Protocol (RTP) to the calling VoIP

5   phone. For this example, the caller's and message robot RTP functions set the DSCP value of

the voice packets to 45 for Priority precedence.

At step 6, the voice conversation has ended and the recipient terminates the connection.

The recipient's User Agent sends a BYE message to the caller. For purposes of this example, the

message robot's user agent sent the BYE message displayed in Figure 11 to the SIP Proxy at IP

10   address 192.168.1.234 and port 5060 with the packet DSCP value set to 45.

The SIP Proxy then sets the DSCP value to 45 for a priority packet and forwards the BYE

message to the caller's user agent at 192.168.1.102 and port 5060 as shown in Figure 12.

At step 7, the caller's User Agent sends a "200 OK" response to the recipient (message

robot) to acknowledge the call has ended. For purposes of this example, the caller's user agent

15   set the "200 OK" packet DSCP value to 45 and sent this message to the SIP Proxy at IP address

192.168.1.234 and port 5060. The "200 OK" message from the caller's user agent to the SIP

Proxy is shown in Figure 13.

The SIP Proxy then sets the DSCP value to 45 for a Priority packet and forwards the "200

OK" response to the message robot at 192.168.1.234 and port 5001 as shown in Figure 14.

20   An exemplary embodiment of the system of the instant invention may employ Linux

traffic control to implement the MLEF PHB. The Linux 2.4.x kernel supports many routing and

packet classifying functions. A personal computer may be used with two Ethernet network

interface cards (NIC) as a Linux router to route packets between two local area networks

(LANs). One NIC supports FastEthernet and the second NIC implements 10Mbps 10Base-T

Ethernet. Again, such configuration is exemplary only of a system implementing the invention,

as the MLEF PHB is operating system independent and may, in fact, be deployed in any router.

Figures 15, 16, and 17 illustrate how the kernel receives and forwards TCP/IP packets.

5   These are steps taken by the kernel to receive an incoming packet and route it to the destination

host:

(1)   A packet arrives at the network interface card.

(2)   The network interface card triggers an interrupt.

(3)   The kernel creates a sk_buff data structure to receive the incoming packet from

10   the network interface card.

(4)   The kernel adds the sk_buff packet to the backlog queue. There is only one

backlog queue for the system. If the backlog queue reaches maximum capacity, newly arriving

packets will be dropped.

(5)   The ip_forward() function is called and checks if the packet's time-to-live (i.e.,

15   the number of router hops the packet can traverse before a packet is discarded, marked in the

packet by the sending computer) has expired. If the time-to-live value expires, the packet is

discarded.

(6)   A new packet is constructed, consisting of the contents of the old packet. If the

incoming packet is a fragment, it is placed in a queue waiting for assembly. When all fragments

20   have been received, they are combined into a single sk_buff data structure. There is a timer for

this assembly. If the timer expires, then the packet is marked as lost.

(7)   The kernel IP layer calls the IP routing functions to determine which route to be

taken. This information includes the source IP address, the network device data structure,

maximum message size, a reference count, a usage count, and a last usage timestamp.

(8)   When an IP route is looked up, the route cache is first checked for a matching

5   route.

(9)   If a route is not found in the route cache, the kernel's routing table is queried.

(10)   If the next route is found in the kernel's routing table, the route cache is updated.

(11)   The ip_queue_xmit function is called to place the outgoing packet in the wait

queue.

10   (12)   The network device driver is activated for transmission. The queuing algorithm

for a network interface card determines if the packet should be transmitted, held in the queuing

discipline's queue, or dropped.

The MLEF enqueuing operations are illustrated in Figure 16. The MLEF code

determines the Type of Service to be used from the packet, by checking the DSCP. The DSCP

15   comes from the DiffServ value, which is part of the ToS byte.

(13)   First, the packet is checked to determine if it should be given a particular Type of

Service. Identification of a packet as a voice call with precedence is determined by matching the

packet's DiffServ value with preset DiffServ values for each precedence level. The DiffServ

values for each call precedence are shown in the table of Figure 18. If the incoming packet has a

20   DiffServ value that matches the DiffServ value that has been assigned for a call precedence, the

packet is evaluated to be queued in the priority queue. Otherwise, the packet will be queued in

the best effort queue.

(14)   The MLEF algorithm is implemented within the priority queue for the transmit

functions of the network interface card. There is one priority queue and there can be multiple

other non-priority queues in the transmit functions of the router network interface. For purposes

of illustration only, the exemplary queues described herein are the priority queue and the best

5   effort queue. The priority queue is always dequeued before the best effort queue. The number

of packets in the MLEF queue cannot exceed the maximum transmit queue length (set to 100

during testing).

In MLEF, within the priority queue, each call precedence has a maximum number of

packets that can be queued as shown in the table of Figure 18. If a call packet arrives and the

10   number of packets for the respective Type of Service, as determined by its DiffServ value, is at

maximum, the incoming packet is dropped.

(15)   After the incoming packet is identified for the priority queue, the kernel evaluates

if the maximum transmit queue length has been reached. If so, the incoming packet is dropped.

(16)   If the maximum transmit queue length has not been reached, the voice packet is

15   enqueued in the priority queue. The number of packets within the priority queue is incremented

and the number of packets within the transmit queue is incremented.

(17)   If the incoming packet did not have a DSCP value indicating that the packet

should have the Type of Service that has been assigned for the priority queue, the packet will be

sent to the best effort queue. If the transmit queue occupancy for the particular network card

20   reaches the maximum value, the packet is dropped. Otherwise, the packet will be queued in the

best effort queue.

(18)   The packet is queued in the best effort queue (which is dequeued if the priority

queue is empty). The number of packets in the transmit queue is incremented.

(19)   A packet is dropped if the transmit queue occupancy reaches the maximum value. In addition, a voice packet designated for the priority queue is dropped if the queue size for its DSCP exceeds the preset maximum value. Since call precedences of higher importance have a larger maximum queue size, packets of lower precedence calls will be dropped before packets of

5   higher precedence.

(20)   When the kernel is ready to dequeue a packet and transmit the queued packets, the events in Figure 17 occur. In the system used for testing, all the packets in the transmit buffer queues may not be transmitted before more packets are added to the transmit buffer, due to the interrupt nature of the test system.

10   (21)   The priority queue is checked first for any packets to transmit.

(22)   If the priority queue does have a packet, the packet is dequeued from the priority transmit queue. In the exemplary embodiment described herein, the packets may be dequeued in First In First Out (FIFO) order.

(23)   The ToS value is obtained from the packet to be dequeued. Since the entire

15   transmit buffer may not be emptied, the system keeps track of the priority transmit queue length for each precedence level, separately.

(24)   The priority queue length for the packet's ToS value is decremented. The transmit queue length is decremented.

(25)   The packet to be transmitted is sent to the network interface card.

20   (26)   The network interface card transmits the packet to the destination host over the physical media (e.g., an Ethernet cable).

(27)   If the priority queue is empty when the kernel is ready to transmit a packet, the best effort queue is checked for packets to be sent.

(28)	The packet is dequeued from the best effort queue.

(29)	The transmit buffer queue length is decremented and the packet is sent to the network interface card for transmission.

(30)	The kernel continues processing.

5	An exemplary hardware system employing the instant invention is shown in Figure 19, and comprises a Linux router configured to send packets to a receiving application in a 10Base-T Ethernet LAN. The MLEF algorithm was implemented in the Linux kernel for the output of the 10Base-T Ethernet card. This configuration simulated a SIP phone call on a congested network via the MLEF Linux router between the local area networks. The SIP user agents participating

10	in the phone call operated on separate computers from the computers congesting the network.

The Fast Ethernet interface on the MLEF Linux router served as a gateway for an isolated local area network (LAN). The FastEthernet LAN serviced three Linux workstations. Two workstations served as network load generator and the third workstation operated as a Session Initiation Protocol (SIP) Proxy and message robot.

15	The SIP Proxy was an open source SIP Proxy project that was modified to parse the precedence of the SIP call from the SIP request message. The SIP Proxy marked the Type of Service (ToS) value on SIP responses sent to the message robot and to the calling SIP user agent corresponding to the precedence of the SIP call request message from the calling user agent.

The message robot receiver consisted of an open source SIP user agent that streamed a

20	pre-recorded message to the SIP user agent that called the message robot. The message robot set the Type of Service (ToS) byte on the packets corresponding to the precedence requested from the calling SIP user agent. Using a set voice message to stream to the calling SIP user agent

served as a reference for comparing the effects of network congestion on the quality of speech in the phone call.

The load generator computer congested the network by creating multiple User Datagram Protocol (UDP) streams and transmitting them to the load receiver computer. The UDP streams consisted of streaming 200-byte packets at 1000 packets per second to separate destination ports on the load-receiving computer. The load receiver computer connected to the 10Base-T LAN.

The 10Base-T network interface in the MLEF Linux router served as the gateway for the load receiver computer and a SIP user agent. The SIP user agent provided call signaling for a VoIP softphone. The SIP user agent and VoIP softphone were bundled together in an open source project. The SIP user agent was modified to set the ToS value in SIP messages corresponding to the precedence of the call.

The load-receiving computer operated a UDP listening application to receive the UDP packets sent by the load generator computer.

The SIP Phone used G.711 coder-decoder (codec) for VoIP calls with a 20 ms sampling time, generating 50 packets per second. G.711 uses 64 kb/s. Each packet contains UDP, RTP, and an IP header, a 40-byte header. Forty bytes times fifty packets per second yields 16 kb/s in packet overhead. This results in a data stream of 80 kb/s with no silence suppression.

Measurements were taken of the number of packets for each call precedence that were received, transmitted, and dropped by the MLEF code within the packet forwarding processes of the 10Base-T network interface card in the Linux router. The measurements demonstrated that during network congestion, voice packets of higher precedence were not dropped while voice packets of lower precedence were dropped. This is the desired behavior for a router to ensure that high priority VoIP calls can be successfully established with call control packets and voice

conversation is intelligible. The table depicted in Figure 20 contains metrics indicating the level of congestion caused by a higher volume of routine voice packets traversing the router while a phone call is established with a Priority precedence.

While the above description of an exemplary embodiment of a hardware system

5   incorporating the invention particularly notes the use of a Linux router, it should be noted that routers provided with alternate operating systems may likewise be used (e.g., UNIX, a specialized operating system built for real-time communications, etc.) without departing from the spirit and scope of the invention. In order to implement the MLEF PHB in such alternate routers, code is inserted into the router operating system to implement the functions set forth

10   above. The MLEF PHB may likewise be implemented in WINDOWS by coding the MLEF process in the network driver software for the Ethernet card.

The invention has been described with references to a preferred embodiment. While specific values, relationships, materials and steps have been set forth for purposes of describing concepts of the invention, it will be appreciated by persons skilled in the art that numerous

15   variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the spirit or scope of the basic concepts and operating principles of the invention as broadly described. It should be recognized that, in the light of the above teachings, those skilled in the art can modify those specifics without departing from the invention taught herein. Having now fully set forth the preferred embodiments and certain

20   modifications of the concept underlying the present invention, various other embodiments as well as certain variations and modifications of the embodiments herein shown and described will obviously occur to those skilled in the art upon becoming familiar with said underlying concept. It is intended to include all such modifications, alternatives and other embodiments insofar as

they come within the scope of the appended claims or equivalents thereof. It should be understood, therefore, that the invention may be practiced otherwise than as specifically set forth herein. Consequently, the present embodiments are to be considered in all respects as illustrative and not restrictive.